

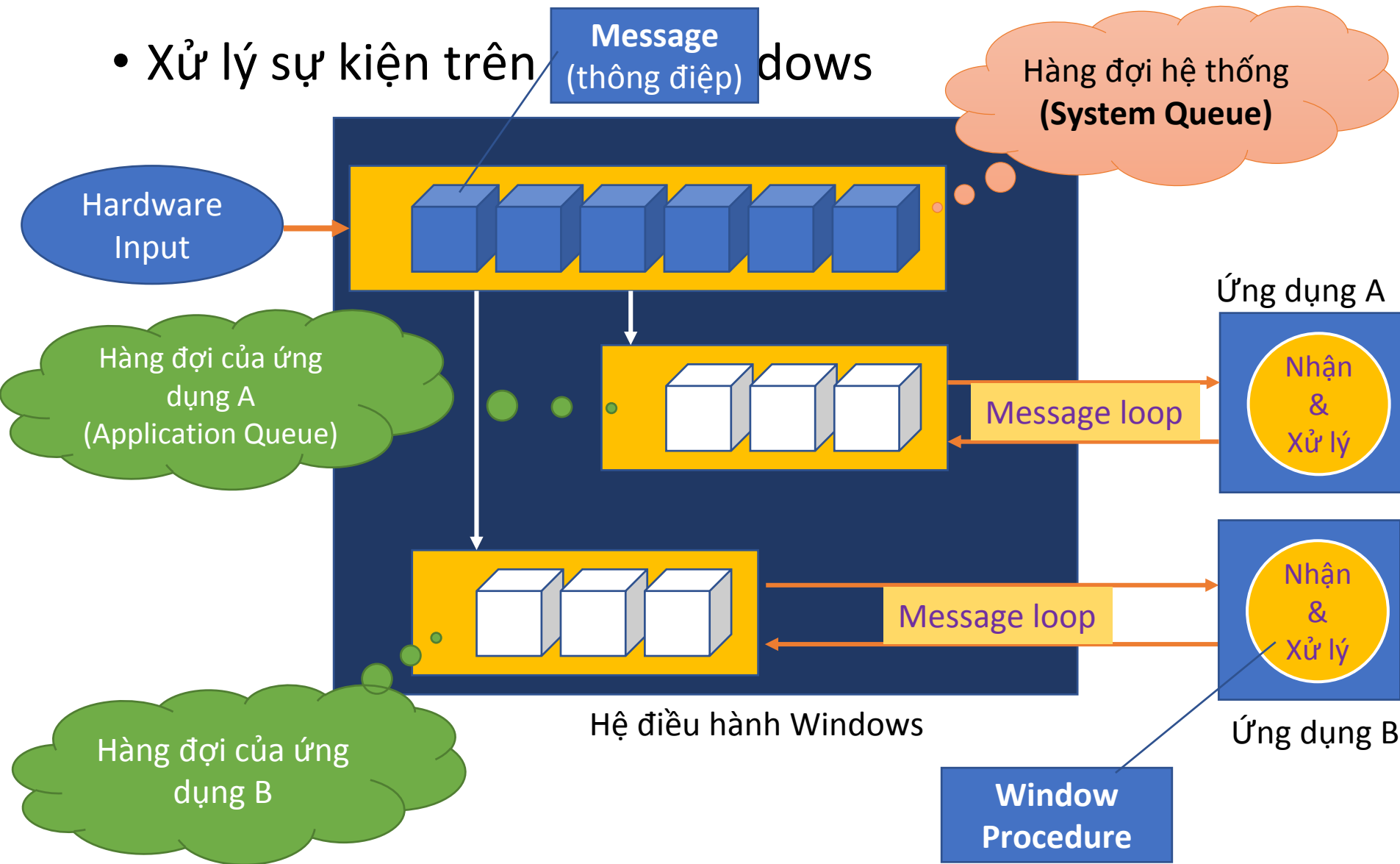
**LẬP TRÌNH WINDOW**

# NỘI DUNG

- Lập trình xử lý sự kiện
- Lập trình ứng dụng Window Form
- Một số điều khiển cơ bản
- Giao diện MDI
- Lập trình cơ sở dữ liệu

# LẬP TRÌNH XỬ LÝ SỰ KIỆN

- Xử lý sự kiện trên Message windows



# LẬP TRÌNH XỬ LÝ SỰ KIỆN

- **Message (Thông điệp)**

- Một message là một con số nguyên được quy ước trước giữa Windows và các ứng dụng (Application)
- Các dữ liệu nhập (từ bàn phím, từ chuột, ...) đều được Windows chuyển thành các message và một số thông tin kèm theo message
- Ví dụ:
  - 0x0001          WM\_CREATE
  - 0x0002          WM\_DESTROY
  - 0x0003          WM\_MOVE
  - 0x0005          WM\_SIZE
  - 0x0012          WM\_QUIT

# LẬP TRÌNH XỬ LÝ SỰ KIỆN

- **System Queue (Hàng đợi hệ thống):**
  - Hàng đợi để Windows chứa các message.
- **Application Queue (Hàng đợi ứng dụng ):**
  - Hàng đợi riêng của các ứng dụng để chứa các message của ứng dụng.

→ Windows sẽ tự động phân bổ các message từ System Queue đến các Application Queue

- **Message loop (vòng lặp thông điệp)**
  - Mỗi ứng dụng tại một thời điểm có một message loop để lấy các message trong Application Queue về để phân bổ cho các cửa sổ (Window) trong Application

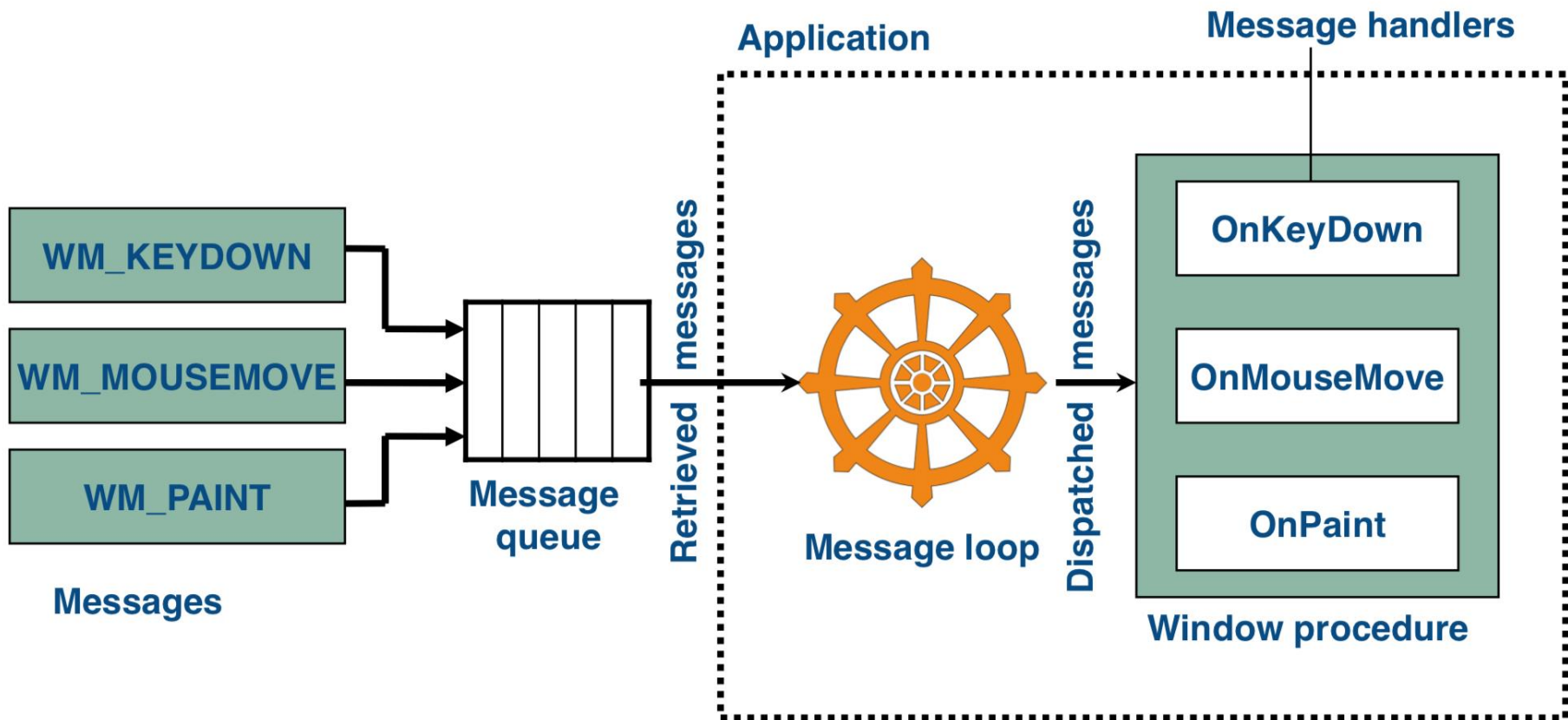
# LẬP TRÌNH XỬ LÝ SỰ KIỆN

- **Hàm Window Procedure**

- Mỗi cửa sổ (Window) trong Application đều có một hàm Window procedure để xử lý các message do message loop nhận về

# LẬP TRÌNH XỬ LÝ SỰ KIỆN

- Mô hình lập trình xử lý sự kiện



# LẬP TRÌNH XỬ LÝ SỰ KIỆN

- Event-driven programming model:
  - Ứng dụng phản ứng các sự kiện (nhấn phím, click chuột, ...) bằng cách xử lý các message do Windows gửi đến
  - Một ứng dụng Windows điển hình thực hiện một lượng lớn các xử lý để phản hồi các message nó nhận. Và giữa các message nó chờ message kế tiếp đến
- Message queue: Các message được chờ trong message queue cho đến khi chúng được nhận để xử lý



# LẬP TRÌNH XỬ LÝ SỰ KIỆN

- Hàm Main: tạo một cửa sổ và vào message loop
- Message loop:
  - Nhận các message và phân bổ chúng đến Window Procedure của các cửa sổ
  - Message loop kết thúc khi nhận được WM\_QUIT (chọn Exit từ menu File, click lên close button)
- Window Procedure:
  - Phần lớn các đoạn mã đặt trong Window Procedure.
  - Window Procedure xử lý các message gửi đến cửa sổ
  - Window Procedure điển hình chứa câu lệnh switch lớn với mỗi case là một message riêng.
- Message handler: Code cung cấp để xử lý message cụ thể

# LẬP TRÌNH XỬ LÝ SỰ KIỆN

- Event – driven programming model trong C#
  - Message Loop --> Application.Run()
  - Window --> Form
  - Window Procedure --> WndProc(ref Message m)
  - Phần lớn các Message handlers được cài đặt sẵn trong các lớp có thể nhận message (Control, Form, Timer, ...) dưới dạng các hàm protected:  
**protected void OnTenMessage(**xxxEventArgs** e)**
    - **xxxEventArgs** có thể là EventArgs hay các lớp con của EventArgs
  - Mỗi message có một biến event tương ứng.
  - Các Message handlers mặc nhiên gọi các event tương ứng của message
  - Các hàm gán cho event gọi là event handler

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

1

- Thiết kế giao diện

2

- Xử lý các message do Windows gửi đến

3

- Xử lý nghiệp vụ

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- Ứng dụng Window Form có 3 phần chính:
  - Application
  - Các Form trong Application
  - Các Controls và Components trên Form

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Lớp Application:** cung cấp các phương thức tính và các property tính để quản lý ứng dụng
  - Các phương thức start, stop ứng dụng, xử lý Windows messages,
  - Các property lấy thông tin về ứng dụng
  - Lớp này không thể thừa kế
- Namespace
  - System.Windows.Form
- Assembly
  - System.Windows.Form (System.Windows.Form.dll)

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Lớp Application:** Một số phương thức thông dụng
  - **Run(Form)** bắt đầu message loop của ứng dụng
  - **Exit()** dừng message loop
  - **DoEvents()** xử lý các message trong khi chương trình đang trong vòng lặp
  - **EnableVisualStyles()** các control sẽ vẽ với kiểu visual nếu control và hệ điều hành hỗ trợ
  - **Restart()** dừng ứng dụng và Tự động restart lại

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Lớp Application:** một số property thông dụng
  - **ExecutablePath** Đường dẫn đến file .exe
  - **StartupPath** Đường dẫn đến thư mục chứa file .exe
  - **UseWaitCursor** Hiện cursor dạng Wait
- Event thông dụng:
  - **Idle** Xuất hiện khi ứng dụng hoàn thành việc xử lý

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

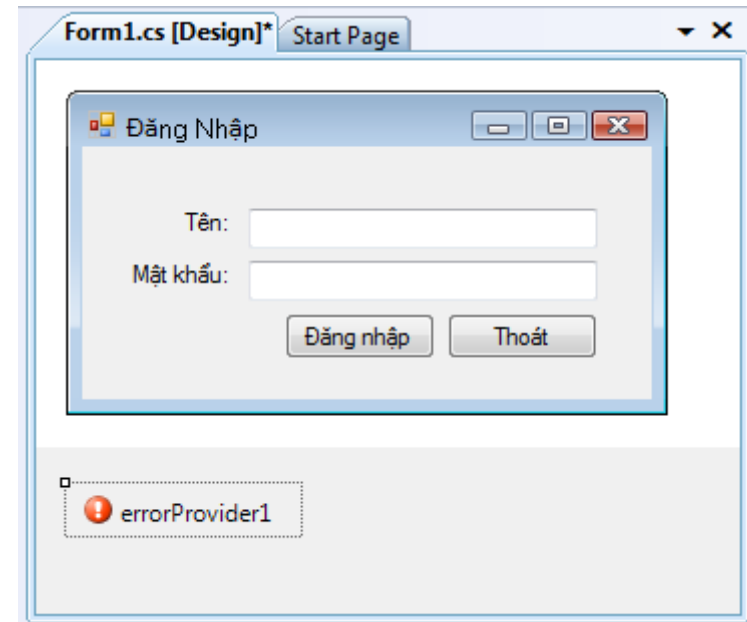
- **Component** là một thành phần phần mềm
- Lớp **Component** là lớp cơ sở của tất cả các component
  - Namespace:
    - System.ComponentModel
  - Assembly
    - System (System.dll)



# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- Các component trong .NET gồm có các thành viên:
  - Properties (các thuộc tính)
  - Methods (các phương thức)
  - Events (các sự kiện)
  - [Các thành viên protected]
- Các Component không hỗ trợ tương tác với người dùng bằng form giao diện tự nhiên

Nonvisual design  
surface



# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- Sơ đồ thừa kế

**System.Object**

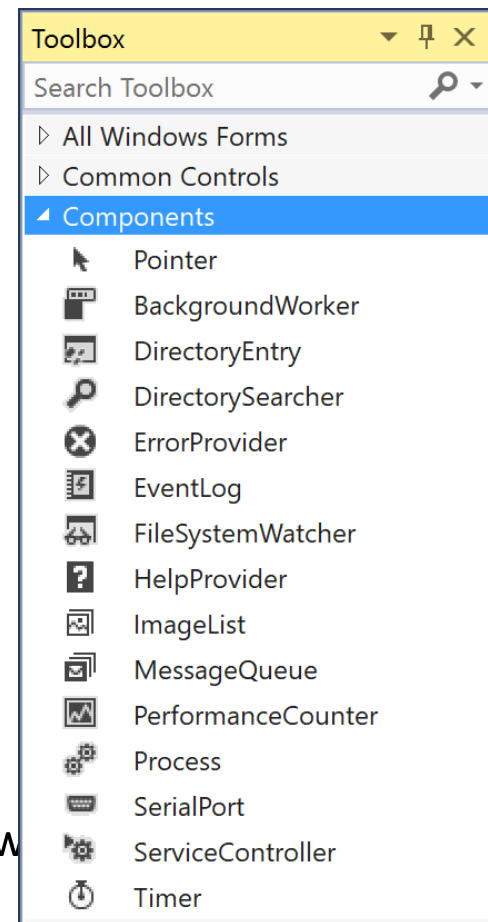
└─ **System.MarshalByRefObject**

└─ **System.ComponentModel.Component**

└─ System.Data.Common.DbConnection  
└─ System.Diagnostics.PerformanceCounter  
└─ System.Diagnostics.Process  
└─ System.Timers.Timer

**System.Windows.Forms.Control**

└─ System.Windows.Forms.ErrorProvider  
└─ System.Windows.Forms.HelpProvider  
└─ System.Windows.Forms.ImageList  
└─ System.Windows.Forms.Menu  
└─ System.Windows.Forms.NotifyIcon  
└─ System.Windows.Forms.StatusBarPanel  
└─ System.Windows.Forms.Timer  
└─ System.Windows.Forms.ToolStripButton  
└─ System.Windows.Forms.ToolStripItem  
└─ System.Windows.Forms.ToolStripPanelRow  
└─ System.Windows.Forms.ToolTip



# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Control** là Component có giao diện
  - Giao diện ứng dụng gồm một tập các control, giúp người dùng tương tác với ứng dụng
  - Cho phép hiển thị dữ liệu (output) hay cho phép nhập dữ liệu vào (input)
- Lớp **Control** là lớp cơ sở cho các control
  - Namespace
    - System.Windows.Forms
  - Assembly
    - System.Windows.Forms (System.Windows.Forms.dll)

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- Sơ đồ thừa kế

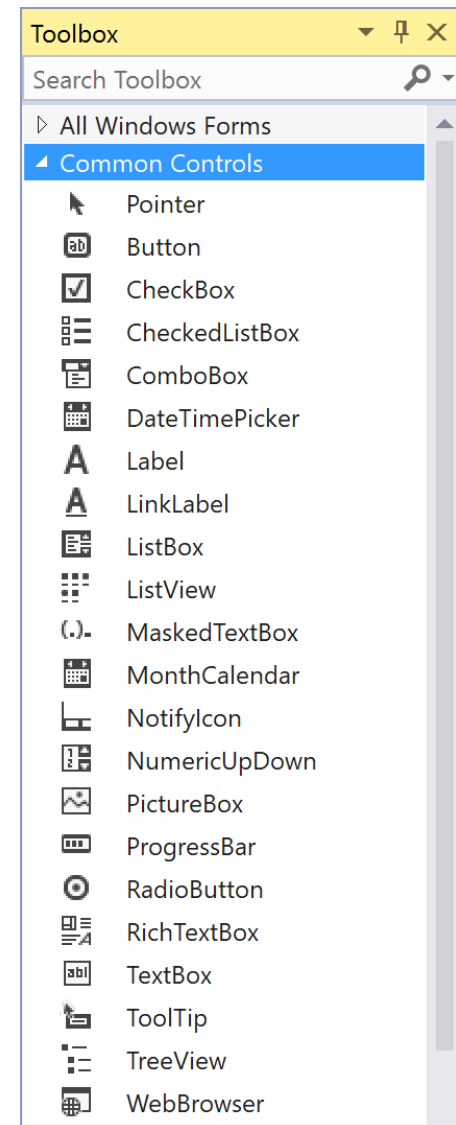
**System.Object**

**System.MarshalByRefObject**

**System.ComponentModel.Component**

**System.Windows.Forms.Control**

System.Windows.Forms.ButtonBase  
System.Windows.Forms.DataGrid  
System.Windows.Forms.DataGridView  
System.Windows.Forms.DateTimePicker  
System.Windows.Forms.GroupBox  
System.Windows.Forms.Label  
System.Windows.Forms.ListControl  
System.Windows.Forms.ListView  
System.Windows.Forms.MdiClient  
System.Windows.Forms.MonthCalendar  
System.Windows.Forms.PictureBox  
System.Windows.Forms.PrintPreviewControl  
System.Windows.Forms.ProgressBar  
System.Windows.Forms.ScrollableControl  
System.Windows.Forms.ScrollBar  
System.Windows.Forms.Splitter  
System.Windows.Forms.StatusBar  
System.Windows.Forms.TabControl  
System.Windows.Forms.TextBoxBase  
System.Windows.Forms.ToolBar  
System.Windows.Forms.TrackBar  
System.Windows.Forms.TreeView  
System.Windows.Forms.WebBrowserBase

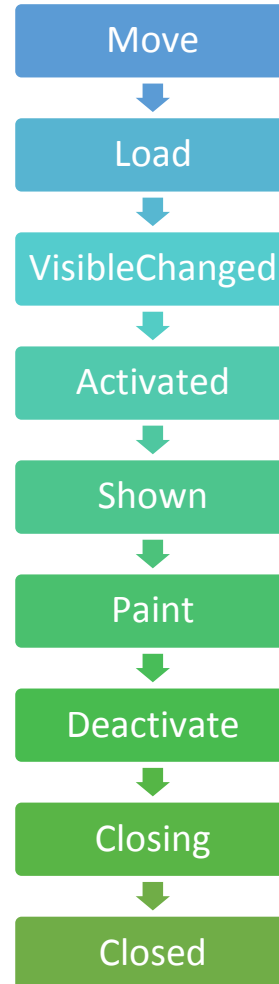


# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- Lớp **Form** thể hiện một cửa sổ (window) hay một dialog box tạo nên giao diện của ứng dụng
- Thông thường tạo custom form bằng cách thừa kế từ lớp Form
- Namespace
  - System.Windows.Form
- Assembly
  - System.Windows.Form (System.Windows.Form.dll)

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- Lớp **Form**
  - Chu trình đời sống của form



# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- Lớp Form

Properties

Form1 System.Windows.Forms.Form

**Accessibility**

AccessibleDescription	
AccessibleName	
AccessibleRole	Default

**Appearance**

BackColor	Control
BackgroundImage	(none)
BackgroundImageLayout	Tile
Cursor	Default

**Font**

Font	Microsoft Sans Serif, 7.875pt
ForeColor	ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
<b>Text</b>	<b>Form1</b>
UseWaitCursor	False

**Behavior**

AllowDrop	False
AutoValidate	EnablePreventFocusChange
ContextMenuStrip	(none)
DoubleBuffered	False
Enabled	True
ImeMode	NoControl

**Data**

(ApplicationSettings)	
(DataBindings)	
Tag	

**Design**

(Name)	<b>Form1</b>
Language	(Default)
Localizable	False
Locked	False

Properties

Form1 System.Windows.Forms.Form

**Focus**

CausesValidation	True
------------------	------

**Layout**

AutoScaleMode	<b>Font</b>
AutoScroll	False
AutoScrollMargin	0, 0
AutoScrollMinSize	0, 0
AutoSize	False
AutoSizeMode	GrowOnly
Location	0, 0
MaximumSize	0, 0
MinimumSize	0, 0
Padding	0, 0, 0, 0
Size	<b>834, 494</b>
StartPosition	WindowsDefaultLocation
WindowState	Normal

**Misc**

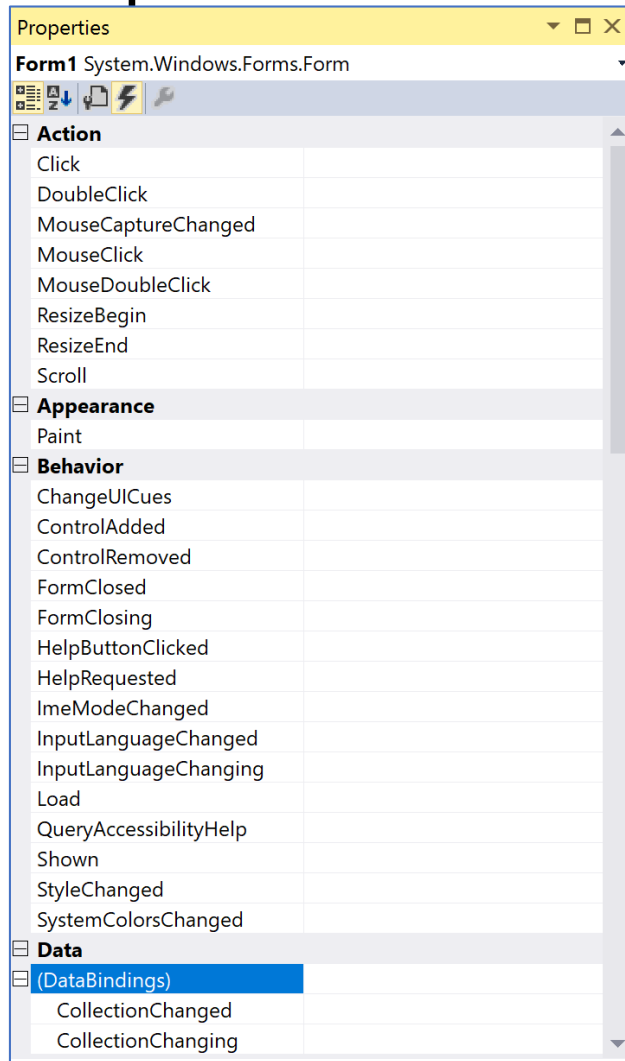
AcceptButton	(none)
CancelButton	(none)
KeyPreview	False

**Window Style**

ControlBox	True
HelpButton	False
Icon	(Icon)
IsMdiContainer	False
MainMenuStrip	(none)
MaximizeBox	True
MinimizeBox	True
Opacity	100%
ShowIcon	True
ShowInTaskbar	True
SizeGripStyle	Auto
TopMost	False
TransparencyKey	

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

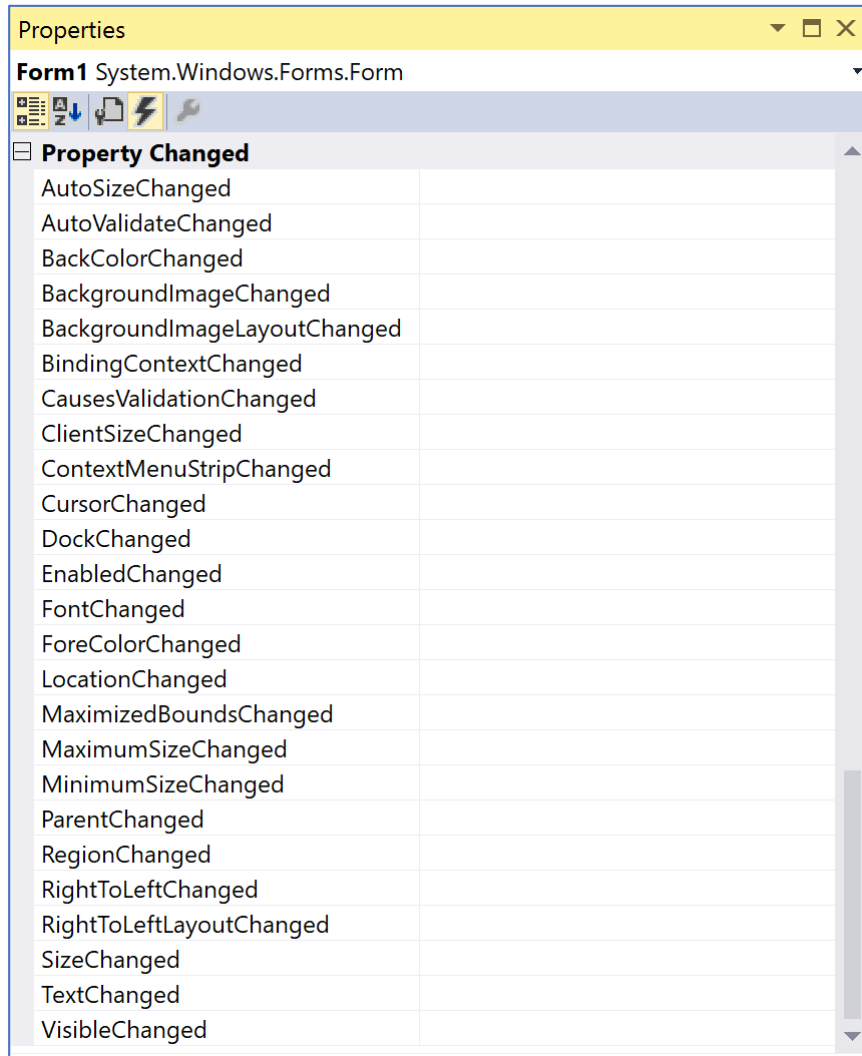
## • Lớp Form





# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- Lớp Form



# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Label** control dùng để cung cấp chuỗi mô tả cho control
  - Một số property thông dụng
    - **Text**, TextAlign, Image, ImageAlign, Visible
    - BackColor, ForeColor
    - Font
  - Một số phương thức thông dụng
    - Hide(), Show()
  - Một số event thông dụng
    - Paint

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **TextBox** control dùng để nhập chuỗi vào ứng dụng
  - Cho phép nhập nhiều dòng
  - Tạo mặt nạ để nhập password
  - Một số property thông dụng
    - **Text**, CharacterCasting, ReadOnly, PasswordChar (Multiline=false), MaxLength
    - Multiline, ScrollBars, WordWrap, Lines[], AcceptTab, AcceptReturn
  - Một số phương thức thông dụng
    - Clear(), Cut(), Paste(), Copy(), Undo(), Select(), SelectAll(), DeselectAll(), ProcessCmdKey()
  - Một số event thông dụng
    - Click, Enter, Leave, **TextChanged**, MultilineChanged, **KeyPress**

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Button** control cho phép người dùng click lên nó để thực hiện một hành động
  - Một số property thông dụng
    - Text, Image, TextAlign, ImageAlign, DialogResult
  - Một số phương thức thông dụng
    - PerformClick()
  - Một số event thông dụng:
    - Click, MouseEnter, MouseLeave

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Message Box** hiện một thông báo hay một hướng dẫn cho user
- Lớp **MessageBox** chỉ chứa một phương thức tĩnh duy nhất: **Show(...)**

```
DialogResult Show(string text, string caption,  
                  MessageBoxButtons buttons,  
                  MessageBoxIcon icon,  
                  MessageBoxDefaultButton defaultButton,  
                  MessageBoxOptions options);
```

- Namespace:
  - System.Windows.Forms
- Assembly
  - System.Windows.Forms(System.Windows.Forms.dll)

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

## • Lớp MessageBox

```
public enum MessageBoxButtons
{
    OK,
    OKCancel,
    AbortRetryIgnore,
    YesNoCancel,
    YesNo,
    RetryCancel
}
```

```
public enum MessageBoxOptions
{
    DefaultDesktopOnly = 0x20000,
    RightAlign = 0x80000,
    RtlReading = 0x100000,
    ServiceNotification = 0x200000
}
```

```
public enum MessageBoxDefaultButton
{
    Button1 = 0,
    Button2 = 0x100,
    Button3 = 0x200
}
```

```
public enum MessageBoxIcon
{
    Asterisk = 0x40,
    Error = 0x10,
    Exclamation = 0x30,
    Hand = 0x10,
    Information = 0x40,
    None = 0,
    Question = 0x20,
    Stop = 0x10,
    Warning = 0x30
}
```

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

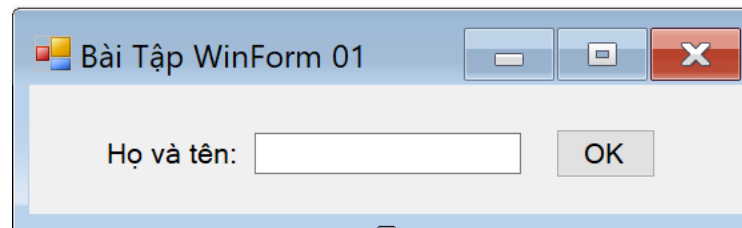
- Lớp **MessageBox**

```
public enum DialogResult
{
    None,
    OK,
    Cancel,
    Abort,
    Retry,
    Ignore, Yes,
    No
}
```

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **BÀI TẬP 1**

- Thiết kế giao diện chương trình như hình bên dưới



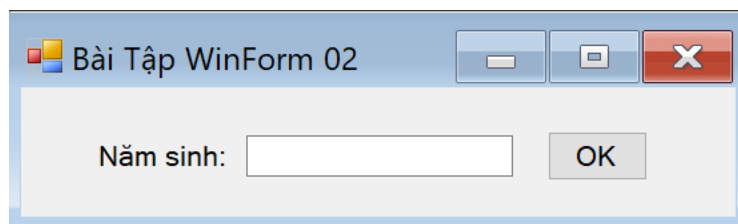
- Nhập **họ và tên** vào Textbox, sau đó nhấn vào Button **OK** thì hiển thị thông báo ***"Xin chào ..."*** lên MessageBox.
- Lưu ý kiểm tra và thông báo lên MessageBox nếu chưa nhập dữ liệu.



# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

## • BÀI TẬP 2

- Thiết kế giao diện chương trình như hình bên dưới

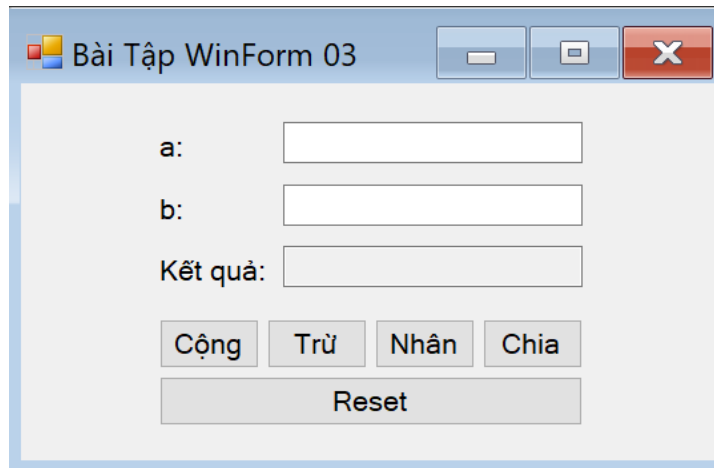


- Nhập **năm sinh** vào Textbox, sau đó nhấn vào Button **OK** thì hiển thị thông báo “**Số tuổi của bạn là: ...**” lên MessageBox.
- Lưu ý kiểm tra và thông báo lên MessageBox nếu chưa nhập dữ liệu.

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **BÀI TẬP 3**

- Thiết kế giao diện chương trình như hình bên dưới



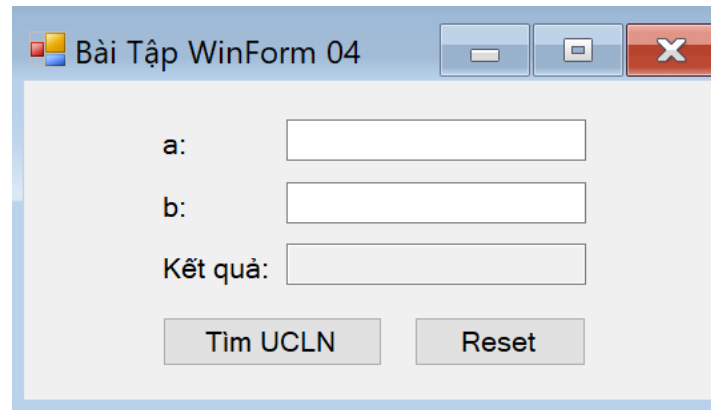
The screenshot shows a Windows application window titled "Bài Tập WinForm 03". Inside the window, there is a light gray background. At the top, there are three text boxes. The first is labeled "a:", the second "b:", and the third "Kết quả:". Below these text boxes, there are five buttons arranged in two rows. The first row contains four buttons: "Cộng", "Trừ", "Nhân", and "Chia". The second row contains a single button labeled "Reset".

- Nhập **2 số nguyên a, b** vào 2 Textbox, sau đó thực hiện các phép tính khi nhấn vào các Button tương ứng **Cộng** ( $a + b$ ), **Trừ** ( $a - b$ ), **Nhân** ( $a * b$ ), **Chia** ( $a / b$ ) và hiển thị kết quả lên Textbox **Kết quả**.
- Nhấn vào nút **Reset** thì xóa tất cả nội dung trong các Textbox.
- Lưu ý kiểm tra và thông báo lên **MessageBox** nếu chưa nhập dữ liệu.

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **BÀI TẬP 4**

- Thiết kế giao diện chương trình như hình bên dưới



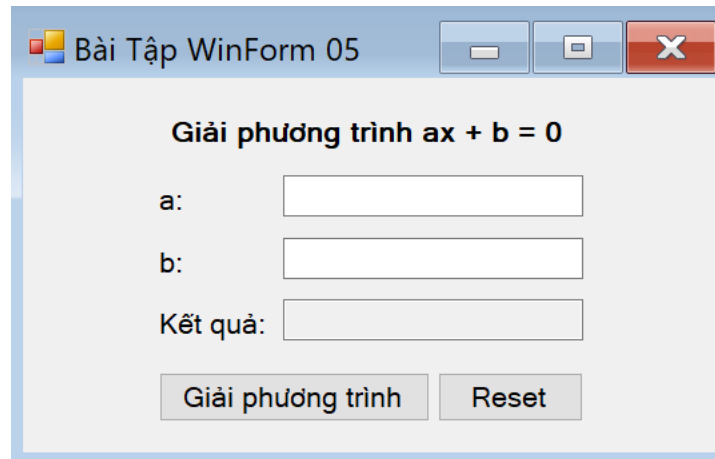
The screenshot shows a standard Windows application window with the title bar 'Bài Tập WinForm 04'. Inside the window, there are three text input fields. The first is labeled 'a:', the second 'b:', and the third 'Kết quả:'. Below these fields, there are two buttons: 'Tìm UCLN' (Find GCD) and 'Reset'. The window has standard minimize, maximize, and close buttons in the title bar.

- Nhập **2 số nguyên a, b** vào 2 Textbox, sau đó nhấn vào Button **Tìm UCLN** thì sẽ hiển thị ước chung lớn nhất của 2 số nguyên a, b vào Textbox **kết quả** .
- Nhấn vào nút **Reset** thì xóa tất cả nội dung trong các Textbox.
- Lưu ý kiểm tra và thông báo lên MessageBox nếu chưa nhập dữ liệu.

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **BÀI TẬP 5**

- Thiết kế giao diện chương trình như hình bên dưới



The screenshot shows a Windows application window titled "Bài Tập WinForm 05". Inside the window, the text "Giải phương trình  $ax + b = 0$ " is displayed. Below this text, there are three input fields: the first is labeled "a:", the second is labeled "b:", and the third is labeled "Kết quả:". At the bottom of the form, there are two buttons: "Giải phương trình" and "Reset".

- Nhập **2 số nguyên a, b** vào 2 Textbox, sau đó nhấn vào Button **Giải phương trình** thì sẽ hiển thị kết quả của việc giải phương trình  $ax + b = 0$  vào Textbox **kết quả**.
- Nhấn vào nút **Reset** thì xóa tất cả nội dung trong các Textbox.
- Lưu ý kiểm tra và thông báo lên MessageBox nếu chưa nhập đủ liệu.

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **BÀI TẬP 6**

- Thiết kế giao diện chương trình như hình bên dưới

The screenshot shows a Windows application window titled "Bài Tập WinForm 06". Inside the window, the text "Giải phương trình  $ax^2 + bx + c = 0$ " is displayed. Below this text, there are three input fields labeled "a:", "b:", and "c:". Below these fields is a label "Kết quả:" followed by an empty input field. At the bottom of the form, there are two buttons: "Giải phương trình" and "Reset".

- Nhập **3 số nguyên a, b** vào 3 Textbox, sau đó nhấn vào Button **Giải phương trình** thì sẽ hiển thị kết quả của việc giải phương trình  $ax^2 + bx + c = 0$  vào Textbox **kết quả**.
- Nhấn vào nút **Reset** thì xóa tất cả nội dung trong các Textbox.
- Lưu ý kiểm tra và thông báo lên MessageBox nếu chưa nhập dữ liệu.

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Lớp Form:**

- Một số phương thức thông dụng
  - Show(), ShowDialog(), Hide(), Close()
  - CenterToScreen(), DrawToBitmap(), Invalidate()
  - CreateGraphic()
- Một số property thông dụng khác (không có trong Design)
  - MdiParent, MdiChilden • DialogResult
  - Controls

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Form:**

- Kiểm tra có muốn đóng Form hay không
  - Xử lý sự kiện Closing
  - Nếu không muốn đóng form: `e.Cancel = true`
- Thứ tự active của các control trên form (TabIndex)
- View → Tab Order

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Form:**

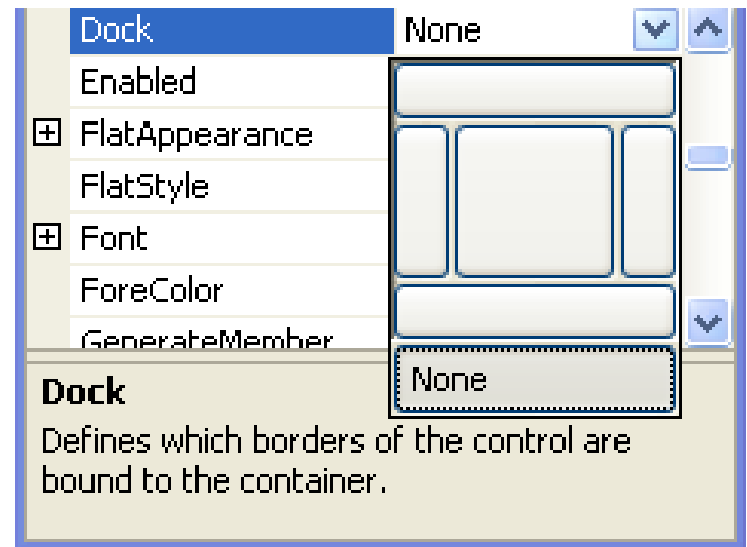
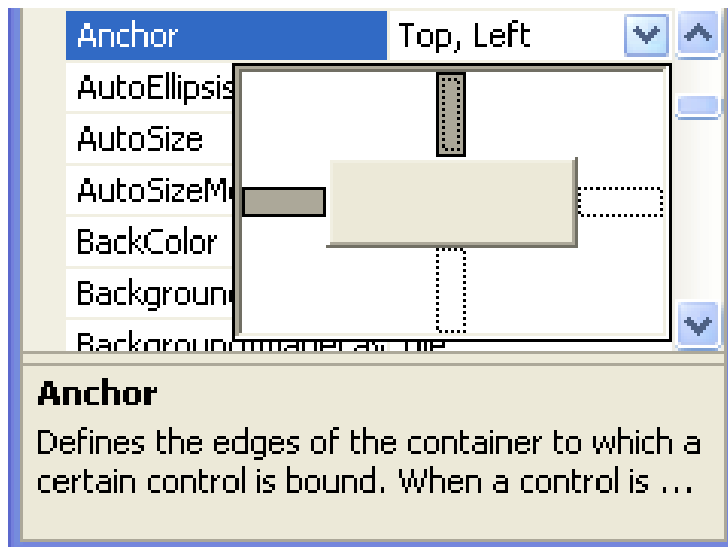
- Bố cục các controls trên form

- Anchor: Chỉ ra các cạnh của container để biết control sẽ thay đổi kích thước như thế nào khi cha nó thay đổi kích thước

- Các cạnh của container: **Left, Top, Right, Bottom**

- Dock: Chỉ ra các cạnh của control sẽ bám vào container

- **Left, Top, Right, Bottom, Fill**





# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Textbox:**

- Chỉ cho phép nhập các ký tự mong muốn
  - Xử lý sự kiện KeyPress
  - Sử dụng thuộc tính KeyChar của tham số sự kiện KeyPressEventArgs
  - Thiết lập giá trị cho thuộc tính Handle của tham số sự kiện KeyPressEventArgs (**Handle = true** → **không hiển thị lên Textbox**)

- **Button:**

- Thiết lập Hotkey:
  - Đặt dấu & trước ký tự muốn đặt Hotkey trong thuộc tính Text
  - Sử dụng Hotkey: ALT + Hotkey

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Checkbox:**



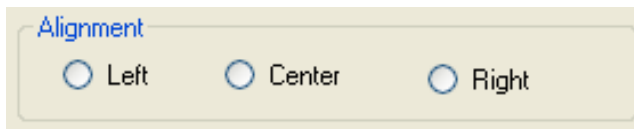
- Một số property thông dụng
  - Text – chữ hiện kế bên checkbox
  - Checked
    - Checked = true → check box đã được check
    - Checked = false → check box chưa được check
- Sự kiện thông dụng
  - CheckedChanged – sự kiện phát sinh khi thay đổi trạng thái check

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Checkbox:** ☐ Multi-Select
  - Một số property thông dụng
    - Text – chữ hiện kế bên checkbox
    - Checked
      - Checked = true → check box đã được check
      - Checked = false → check box chưa được check
  - Sự kiện thông dụng
    - CheckedChanged – sự kiện phát sinh khi thay đổi trạng thái check
  - Các dạng khác của CheckBox
    - ThreeState = true: CheckBox có 3 trạng thái
    - Appearance = Button: CheckBox là một button
  - CheckBox 3 trạng thái
    - Dùng property CheckState để kiểm tra nó có là một trong 3 trạng thái
      - Checked
      - Unchecked
      - Indeterminate

# LẬP TRÌNH ỨNG DỤNG WINDOW FORM

- **Radio buttons:**



- Tương tự checkbox, tuy nhiên các button trong cùng nhóm chỉ có một button được check tại một thời điểm
- Một nhóm: Các radio button được đặt trong cùng container – thường là **panel** hay **group box**
- Property thông dụng:
  - Checked – Cho biết button có được check hay không
- Sự kiện thông dụng:
  - CheckedChanged – Sự kiện phát sinh khi check box được check hay không được check